Efficient Coding of Shape and Transparency for Video Objects

Aghito, Shankar Manuel; Forchhammer, Søren

Published in:
I E E E Transactions on Image Processing

Link to article, DOI:
10.1109/TIP.2007.903902

Publication date:
2007

Document Version
Publisher's PDF, also known as Version of record

Efficient Coding of Shape and Transparency for Video Objects

Shankar Manuel Aghito, *Member, IEEE*, and Søren Forchhammer, *Member, IEEE*

***Abstract*—A novel scheme for coding gray-level alpha planes in object-based video is presented. Gray-level alpha planes convey the shape and the transparency information, which are required for smooth composition of video objects. The algorithm proposed is based on the segmentation of the alpha plane in three layers: binary shape layer, opaque layer, and intermediate layer. Thus, the latter two layers replace the single transparency layer of MPEG-4 Part 2. Different encoding schemes are specifically designed for each layer, utilizing cross-layer correlations to reduce the bit rate. First, the binary shape layer is processed by a novel video shape coder. In intra mode, the DSLSC binary image coder presented in [3] is used. This is extended here with an intermode utilizing temporal redundancies in shape image sequences. Then the opaque layer is compressed by a newly designed scheme which models the strong correlation with the binary shape layer by morphological erosion operations. Finally, three solutions are proposed for coding the intermediate layer. The knowledge of the two previously encoded layers is utilized in order to increase compression efficiency. Experimental results are reported demonstrating that the proposed techniques provide substantial bit rate savings coding shape and transparency when compared to the tools adopted in MPEG-4 Part 2.**

***Index Terms*—Alpha plane, MPEG-4, object-based video, shape coding, transparency coding.**

I. Introduction

In MPEG-4 Part 2 object-based video, the video may be composed of video objects (VOs): each VO is represented by the texture information and the alpha plane. The texture consists of the luminance and the chrominance components. The alpha plane may either be represented by a binary mask, in which case it describes the shape of the VO, or by a gray-level array with values from 0 up to 255, in order to represent both the shape and the transparency level of each pixel in the VO [10], [23]. This description allows for the representation of scenes composed of multiple objects.

The use of VOs provides several advantages compared to traditional video coding techniques. In terms of subjective visual quality, the sharpness of the border of the object may easily be preserved even at low rates, since the binary shape may typically be coded very efficiently even in lossless or near-lossless mode. Selected VOs of interest may be prioritized, e.g., by encoding them with lower quantization factors. Intelligent rate control algorithms may be designed in order to allocate more bits to the desired VOs. In a similar way, in error-prone environments the VOs of interest may be prioritized by using more powerful error resilience. Alternatively, stronger protection could be allocated for the shape [27], and existing techniques for error concealment of corrupted texture information could be improved by using the intact shape information. Shape error concealment was recently investigated [21], [22]. New interactive applications could be conceived exploiting the object-based representation, creating new forms of visual communication. In terms of coding efficiency, benefits from using object-based video were reported in specific applications such as storage of surveillance video [24].

In MPEG-4 Part 2 [10], the binary shape of the VO is extracted from the alpha plane using simple thresholding, and encoded as a separate layer. The texture and the transparency information are then processed only for the pixels located within the coded binary shape.

Shape coding has been studied intensively. Different proposals were evaluated resulting in the adoption of context-based arithmetic encoding (CAE) into MPEG-4 Part 2, as described in [12]. Several alternatives to CAE were then proposed: operational rate distortion (ORD) optimization was utilized to improve the performance of vertex-based methods [12]. The skeleton-based method described in [25] also utilizes ORD optimization, producing bit rate savings in the range 8%–18% compared to CAE. The differential chain code algorithm presented in [15] provided an average bit rate saving of less than 1% in lossless mode compared to CAE. A scheme based on quadtree-based segmentation and adaptive arithmetic encoding was proposed in [28]. A recent overview of shape coding techniques was given in [16].

The digital straight line segments coder (DSLSC) [3] was recently proposed for efficient coding of bilevel images with locally straight edges, e.g., single binary shape images and bilevel layers of digital maps. The DSLSC models the edges as digital straight line segments (DSLS) [18], while standard algorithms like JBIG [8], JBIG-2 [9] and MPEG-4 CAE [10], [12] do not fully exploit the information given by the local straightness of the boundary. In this paper, the DSLSC is extended with an inter mode introducing techniques for exploiting the correlation between consecutive images in a binary shape sequence, providing an efficient alternative to MPEG-4 CAE. A recent rate control analysis of MPEG-4 Part 2 video object coding showed that the shape information has high priority in terms of operational rate-distortion [26]. The new H.264/MPEG-4 Part 10 standard [11] has greatly improved compression of texture (luminance and chrominance) compared to previous standards, including

Manuscript received July 21, 2006; revised June 5, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Tamas Sziranyi.

The authors are with Technical University of Denmark, COM•DTU Department of Communications, Optics and Materials, DTU, Ørsteds Plads 343, 2800 Kgs. Lyngby, Denmark (e-mail: sma@com.dtu.dk; sf@com.dtu.dk).

Digital Object Identifier 10.1109/TIP.2007.903902

MPEG-4 Part 2. Hence, more efficient coding of binary shape (and transparency) is of interest for future object-based video encoders.

Although the use of transparency is desirable for smooth composition of VOs, little documented work has been aimed at efficient coding of the transparency information [1], [17]. In MPEG-4 Part 2, transparency is coded with the same techniques used (and designed) for the texture component, namely the discrete cosine transform but in a shape adaptive (SA-DCT) version [10]. Since the spatial characteristics of texture and transparency information are different [1], [17], it is reasonable to expect improved coding efficiency using an encoder which is specifically targeted at transparency data. In this work, several new strategies for coding the transparency of VOs are presented, including a new segmentation of the gray-level alpha component in three layers: the background layer (formed by all the pixels which are not in the binary shape layer), the opaque layer, and the intermediate layer. The last two layers convey transparency information.

The architecture of the proposed coder for gray-level alpha planes is described in Section II. The DSLSC for coding the binary shape layer as well as the novel extensions for inter coding are described in Section III. The new scheme for coding the opaque layer is presented in Section IV. Three different strategies for coding the intermediate layer are proposed in Section V, one of these introducing a distance map to the background and opaque layers. Experimental results are reported in Section VI demonstrating that the proposed techniques provide high coding gains when compared with the tools adopted in MPEG-4 Part 2.

II. Architecture for Alpha Plane Coding

The composition of video objects is shortly described. Consider a video resolution of N_c columns and N_r rows; given a single frame of a video object, let $\alpha(x,y)$ be the alpha component, with values $0 \leq \alpha \leq 255$ and domain $D = \{1,2,\ldots,N_c\} \times \{1,2,\ldots,N_r\}$; let $Y(x,y)$ and $Y_B(x,y)$ be the corresponding luminance component and the luminance of the background object, respectively. The luminance of the composed scene is given by

$$Y_C(x,y) = Y(x,y)\frac{\alpha(x,y)}{255} + Y_B(x,y)\frac{255-\alpha(x,y)}{255}. \quad (1)$$

In the proposed scheme, the alpha plane $\alpha(x,y)$ is segmented into a background layer (L_0), an opaque layer (L_{255}) and an intermediate layer (L_{int}), as follows:

$$\begin{aligned} L_0 &= \{(x,y) \in D | \alpha(x,y) = 0\} \\ L_{255} &= \{(x,y) \in D | \alpha(x,y) = 255\} \\ L_{\text{int}} &= \{(x,y) \in D | 0 < \alpha(x,y) < 255\}. \end{aligned} \quad (2)$$

Note that the background layer L_0 is the complement of the binary shape layer, indicated as $\overline{L}_0$. The coding flow is depicted in Fig. 1. First, the binary shape is directly encoded with the extended DSLSC. The reconstructed background layer is denoted as $\widetilde{L}_0$, and since lossless coding is used, $\widetilde{L}_0 = L_0$. In order to exploit the correlation between layers, L_{255} is encoded referencing L_0. The reconstructed approximation of the opaque layer

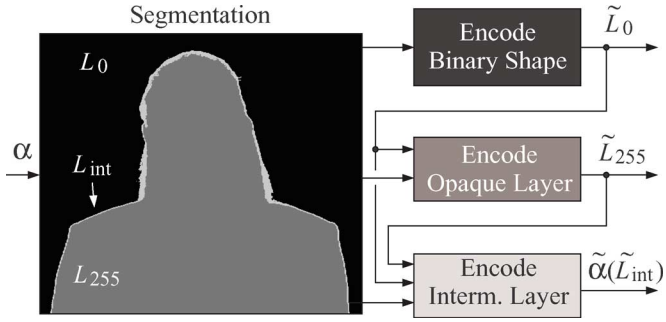


Fig. 1. Proposed architecture.

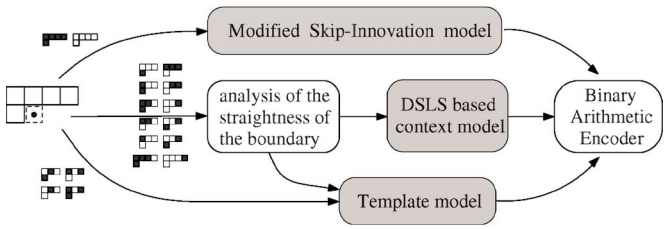


Fig. 2. Architecture of the DSLSC [3], for intra coding of the binary shape.

is indicated as $\widetilde{L}_{255}$. Finally, the transparency values in the intermediate layer $L_{\text{int}} = \overline{L}_0 \setminus \widetilde{L}_{255}$ are encoded referencing both L_0 and $\widetilde{L}_{255}$. The reconstructed values are indicated as $\widetilde{\alpha}(\widetilde{L}_{\text{int}})$.

III. Coding the Binary Shape Layer

The DSLSC intra scheme proposed in [2] and [3] is utilized for coding intra pictures in the binary shape layer $\overline{L}_0$. The scheme is shortly represented below, followed by the description of the novel extensions for exploiting temporal correlation in subsequent binary shape pictures in inter mode.

A. DSLSC for Intra Shape Coding

The DSLSC intra is a lossless context-based arithmetic encoder: causal data is utilized to determine a coding context and pixel values are encoded by arithmetic encoding, sequentially in raster scan order, using distinct probability estimates in each context. The local properties of binary shapes are exploited by utilizing three different models for estimating the probabilities of the pixel being coded: the *DSLS model*, the *modified skip innovation model*, and a simple *template model*. A small template with five causal pixels is used to select the appropriate model, as illustrated in Fig. 2. In order to describe the models, a brief introduction to the concept of chain codes and DSLS is required. For details, refer to [3].

The boundary of a digitized binary shape can be represented as a *chain code*, i.e., a sequence of (unitary length) chain elements, using either the 4-connected or the 8-connected representation. Assuming without loss of generality the first case, let the values 0, 1, 2, and 3 be assigned to chain elements with direction *right*, *up*, *left*, and *down*, respectively. A DSLS is obtained from the digitization of a continuous line segment, e.g., the chain code 0101010101 describes the DSLS produced from the digitization of a line segment forming an angle (close to) $\pi/4$ with the horizontal axis (see also Fig. 3). The properties of DSLSs are well known [3], [7]. The DSLSC utilizes a simple

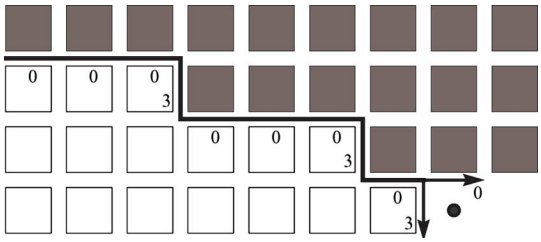


Fig. 3. Example of deterministic decision with the DSLS model. The DSLS 000300030 can only be extended with a 0, since 0003000300 is a DSLS while 0003000303 is not. Assuming that the boundary is straight the chain element has to be 0, and, hence, the pixel to be coded (•) has to be white.

algorithm for parsing/recognizing a DSLS [13], and the concept of *preimage* (in the angle/offset domain) introduced in [7], which given a DSLS, c^l, with l elements, allows to calculate the probability of the next chain element c_{l+1} (under certain conditions) [3].

1) DSLS Model: This is utilized to estimate the probabilities when the pixel to be coded lies along a shape boundary, and the causal portion of the boundary may be described as a DSLS. Note that the pixel value is determined if it is known on which side of the boundary it is located. The DSLSC assumes that the boundary will continue straight, extending the given DSLS (the assumption is often correct in binary shape images). In most cases there is only one possible direction for extending the DSLS, hence only one possible value for the coding pixel (Fig. 3). This value is the prediction produced by the DSLS model. Adaptive context-based arithmetic encoding is utilized to encode whether the prediction is correct or not. The two cases are named *right deterministic decision* (RDD) and *wrong deterministic decision* (WDD), respectively. Typically, RDDs occur more frequently and are, therefore, cheap to code, while WDDs are rare but expensive to code. If the DSLS can be extended in two distinct directions, each inferring a distinct value of the coding pixel, the probability estimates for the two options are calculated using the above mentioned preimage concept (refer to [3] for details).

2) Template Model: This is a simple adaptive template-based model similar to JBIG and MPEG-4 CAE, which is utilized for the irregular portions of the boundary that are not representable as DSLSs, where the DSLS model can not be applied. The size of the template depends on the resolution of the material to encode [3].

3) Modified Skip Innovation Model (MSI): This is utilized to encode runs of consecutive pixels within uniform regions. It is a modification of the *skip innovation* (SI) model utilized in the PWC [5]. The length of the run is initially estimated using the length of the corresponding run on the previous row. If this is terminated by a straight boundary, the direction of the edge is utilized to adjust the initial estimate of when the run stops. If the adjusted estimate is accurate the run is encoded very efficiently. A *failed full skip* (FSS) occurs when the adjusted estimate is greater than the actual run length. These cases typically reduce coding efficiency, since a binary representation of the actual length must be explicitly encoded.

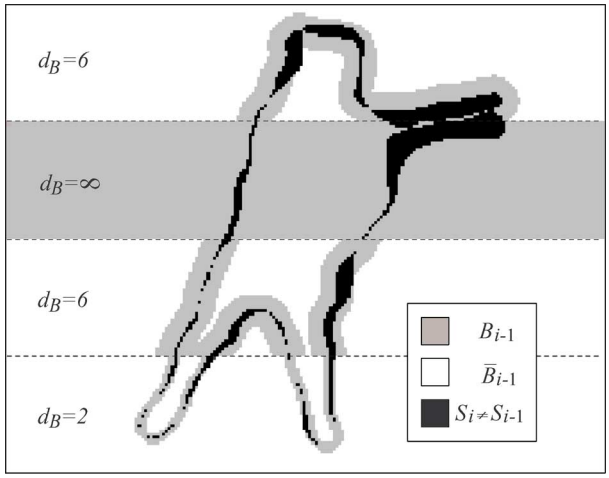


Fig. 4. Reference boundary band B_{i-1} (gray) is constructed around the border of S_{i-1} such that the differences between S_i and S_{i-1} (black) are confined within the band. The width of the band is adjusted in each stripe.

B. Proposed DSLSC Inter Shape Coding

The DSLSC is here extended in order to also exploit temporal redundancies in the shape layer sequences. The extensions are referred to as DSLSC inter and described in the following.

Given the alpha plane α, the binary shape S of a pixel positioned at $p = (x, y)$ is defined as $S(p) = 1$, for $p \in \overline{L}_0$ and $S(p) = 0$, for $p \in L_0$. This can be indicated with the relaxed but intuitive notation $S(\overline{L}_0) = 1, S(L_0) = 0$, which will be used in the rest of this section. For optimal compression of sequences of binary shape images, the correlation between consecutive pictures must be exploited. Let S_i denote the shape of the current picture being encoded, and S_{i-1} denote the shape of the previously encoded reference picture. The reference picture is first utilized to define a band of pixels along the boundary, named *reference boundary band*. The idea is based on the assumption that variations in sequences of binary shapes occur mainly in proximity of the boundary, so that the pixels outside the band are simply copied from the reference picture. It is only the pixels within the band, which are explicitly encoded. For coding these pixels, the DSLSC is modified as described below.

1) Reference Boundary Band: Let B_{i-1} indicate a band of pixels along the boundary of the reference shape, S_{i-1}, defined as follows: a grid point p is in B_{i-1} if and only if the pixels in S_{i-1} within a selected distance d_B from p are not all of identical value, i.e.,

$$B_{i-1} = \{p \in D | \exists a \in D, \exists b \in D | d^E(p,a) < d_B, \; d^E(p,b) < d_B, S_{i-1}(a) \neq S_{i-1}(b)\} \quad (3)$$

where $d^E(\cdot,\cdot)$ is the Euclidean distance between two points and the parameter d_B controls the width of B_{i-1}. Let $\overline{B}_{i-1}$ indicate the points outside the band. Since temporal variations mostly occur along the border of the shape (Fig. 4), there is a high probability $P_{\overline{B}_{i-1}}$ that $S_i(\overline{B}_{i-1}) = S_{i-1}(\overline{B}_{i-1})$. Only one bit is required to signal this event, providing substantial bit savings, since only the pixels $S_i(B_{i-1})$ need to be encoded, while the pixels $S_i(\overline{B}_{i-1})$ are simply copied from S_{i-1}. The larger d_B is,

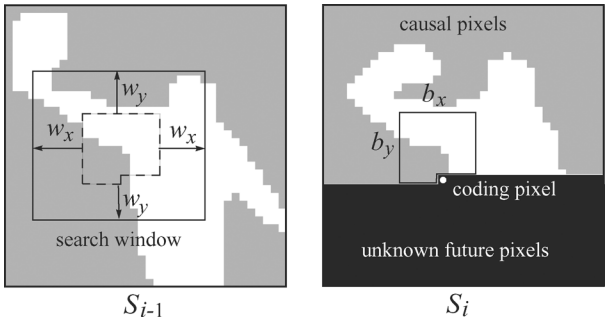


Fig. 5. Motion estimation for the coding pixel is performed utilizing only causal pixels in the current frame, S_i. The size of the block and the search window are determined from the parameters $b_x = b_y = w_x = w_y = 8$.

the higher the probability $P_{\overline{B}_{i-1}}$. On the other hand the width of B_{i-1} has to be small, in order to keep the number of pixels that are explicitly encoded low. Hence, in the current implementation, d_B is chosen as the smallest positive integer (not greater than 7) for which $S_i(\overline{B}_{i-1}) = S_{i-1}(\overline{B}_{i-1})$. Only 3 bits are required to specify d_B or to indicate that the whole binary shape need to be explicitly encoded ($d_B = \infty$). Local adaptivity is obtained partitioning the picture; in the current implementation the picture is divided in stripes, and the procedure described above is applied independently to each stripe, as shown in the example in Fig. 4.

2) Inter Extension of the Template Model: For the pixels $S_i(b), b \in B_{i-1}$ within the band, DSLSC is applied. The coding mode is selected as for the intra case. The Template model is augmented including five pixels in the reference picture S_{i-1}, with coordinates $(x,y), (x+1,y), (x,y+1), (x-1,y), (x,y-1)$, where (x,y) is the location of the pixel being coded in S_i. This is similar to the inter mode of MPEG-4 CAE [12]. If motion occurs between consecutive pictures, it is useful to estimate the motion vector (m_x, m_y) and align the template pixels in S_{i-1} around $(x-m_x, y-m_y)$. Since binary shapes are generally encoded very efficiently by DSLSC even in intra mode, the savings obtained by realigning the template do not necessarily compensate for the cost of coding motion vectors. Hence, a technique that does not introduce additional motion cost is introduced. In the current implementation, motion estimation is performed for each encoded pixel $S_i(x,y)$, evaluating the best match (minimizing the sum of absolute differences) for a block containing *only causal pixels* in the current picture, S_i (Fig. 5). In this way, the decoder can estimate the motion vectors exactly as the encoder does, so they do not need to be encoded in the bitstream. For each horizontal stripe, motion estimation and pixel realignment are only performed when the number of points $b \in B_{i-1}$ for which $S_i(b) \neq S_{i-1}(b)$ is more than a predefined threshold (set to 15% of the total number of points in B_{i-1}). This requires one additional bit for each stripe. An alternative would be to utilize the motion vectors estimated on the luminance component which may be freely available.

More accurate probability estimates and, hence, more efficient coding is achieved exploiting the statistics (counts reported in the different contexts) gathered while encoding the previous pictures. This is done by using the statistics as initialization for the context-based coding process of the current picture.

3) Inter Extension of the DSLS and MSI Models: As mentioned in Section III-A, wrong deterministic decisions and failed full skips are generally expensive. Information collected when the reference picture was coded can be utilized to reduce the frequency and negative impact of these events.

Suppose a pixel $S_i(x,y)$ in proximity of a straight boundary is to be encoded; in intra mode, the DSLS model would be selected. Let W_{i-1} indicate the set of locations of WDDs in the reference picture S_{i-1}. If the Euclidean distance from (x,y) to a point in W_{i-1} is not greater than 2, the (more robust) template model is chosen instead of the DSLS model.

Similarly, suppose a run length starting at $S_i(x,y)$ is being encoded with the MSI model. Let F_{i-1} indicate the location of FFSs (failed full skips) in S_{i-1} (assuming that S_{i-1} was encoded in intra mode). If (x,y) is in F_{i-1} there is a relatively high probability that the FFS would reoccur; hence, the expected length of the run is decreased, reducing the risk of a FFS (if the number of runs of nonsingular elements [3] in the causal DSLS is greater than 3 then the expected length is decreased by 1, otherwise it is decreased by 2).

For the DSLS and MSI models, as for template coding, the statistics gathered while coding previous frames is used to initialize the context-based coding process.

IV. Coding the Opaque Layer

The opaque layer L_{255} is encoded using the knowledge of the background L_0, as illustrated in Fig. 1, exploiting the strong correlation among the two layers which is observed for a large class of alpha planes. The proposed technique consists in representing L_{255} as a morphological erosion of $\overline{L}_0$. The algorithm is organized in a block-based manner, and can be iterated for decreasing block sizes. The strength of the erosion is the key information to be encoded locally for each block. The intra algorithm is presented first, and the modifications for exploiting temporal correlation are given afterward.

A. Block Classification

Blocks that are entirely within L_0 are not processed, since they were encoded by the binary shape coder. These blocks are referred to as *transparent* (TR). Blocks that are entirely in L_{255} are classified as *opaque* (OP). The remaining blocks are classified as *eroded* (ER) or *skipped* (SK), as follows. Local approximations (at block level) of L_{255} are obtained by morphological erosion of $\overline{L}_0$, using filled circles of different size as structuring element. The optimal radius ρ_{opt} of the structuring element is found within a predefined set $R = \{\rho_1, \rho_2, \ldots, \rho_{N_\rho}\}$ (N_ρ is a power of 2), such that the difference between the opaque layer and its lossy representation is minimized. If the approximation error $e(\rho_{\text{opt}})$ is not greater than a preselected threshold $e_{\max}$ and if the alpha values that would be mapped to the approximation of L_{255}, α', are not smaller than a threshold parameter $\alpha_{\min}$, then the block is marked as ER; otherwise, the block is SK. For each block that is not TR, the distance d_{b} from the block to L_0 is defined as the shortest Euclidean distance from pixels in the block to pixels in L_0. If an integer $d_{\text{b}}^\star$ exists, such that all blocks with distance d_{b} not smaller that $d_{\text{b}}^\star$ are OP, then a 0 is coded followed by a binary representation of $d_{\text{b}}^\star$. In this case only blocks with distance $d_{\text{b}} < d_{\text{b}}^\star$ need to be encoded. If $d_{\text{b}}^\star$ does not exist,

TABLE I
DIFFERENT BLOCK TYPES IN THE OPAQUE LAYER CODER AND THE ASSIGNED VARIABLE LENGTH CODES

type	description	C_1	C_2
TR	all pixels in the block are in L_0	-	-
OP	all pixels in the block are in L_{255}	11	-
ER	$\neg$TR $\wedge$ $\neg$OP $\wedge$ $(e(\rho_{\text{opt}}) \leq e_{\max} \wedge \alpha' \geq \alpha_{\min})$	0	0
SK	$\neg$TR $\wedge$ $\neg$OP $\wedge$ $(e(\rho_{\text{opt}}) > e_{\max} \vee \alpha' < \alpha_{\min})$	10	1

TABLE II
VARIABLE LENGTH CODE REPRESENTING Δ_ρ IN PREDICTIVELY ENCODED ER BLOCKS. I$(\cdot,\cdot)$ INDICATES *INNOVATION CODES* [5]

Δ_ρ	code
-1	00
+1	01
0	10
$1 < \Delta_\rho < N_\rho$	11 0 I$(N_\rho - \Delta_\rho - 1, N_\rho - 2)$
$-N_\rho < \Delta_\rho < -1$	11 1 I$(N_\rho + \Delta_\rho - 1, N_\rho - 2)$

a 1 is inserted, indicating that all non TR blocks must be coded. The classification criteria and the variable length codes (VLCs) for each block type are listed in Table I. The code C_1 is used by default, while C_2 is used when in every block there is at least one pixel $p \in L_0$, since in this case there are no OP blocks. After the classification is coded, the different blocks are encoded as follows.

B. Coding Opaque Blocks

Opaque blocks are implicitly encoded in the classification process described above. No additional information is required for these blocks.

C. Coding Eroded Blocks

The optimal radius ρ_{opt} is coded for each ER block. Blocks are processed in raster scan order. A differential scheme is employed. For each ER block the candidate prediction blocks are, in the given order: the block above, the block to the left, and the block in the up-right position. The first available candidate which is ER is selected as a prediction block. If there are no ER candidates, ρ_{opt} is simply coded by a $\log_2(N_\rho)$ bits representation of its index in the ordered set R, indicated as $\mathtt{i}(\rho_{\text{opt}}, R)$. If a prediction block is found, ρ_{opt} is encoded differentially with respect to the optimal radius ρ_p of the prediction block. The difference between the indexes $\Delta_\rho = \mathtt{i}(\rho_{\text{opt}}, R) - \mathtt{i}(\rho_p, R)$, is coded using the VLC defined in Table II.

D. Coding Skipped Blocks

Skipped blocks could not be approximated as an erosion of the shape $\overline{L}_0$. The procedure described above is iterated for decreasing block sizes; each skipped block (Level 1) is divided into four sub-blocks; the sub-blocks are also classified as TR, OP, ER, or SK and coded as described above. Any skipped sub-blocks (Level 2) is divided again, and so on. The iteration stops at a predefined level. The current implementation utilizes four levels, with blocks of size 180×162, 90×81, 45×27, and 45×9. If there are skipped sub-blocks also at the last level, these sub-blocks are encoded with DSLSC.

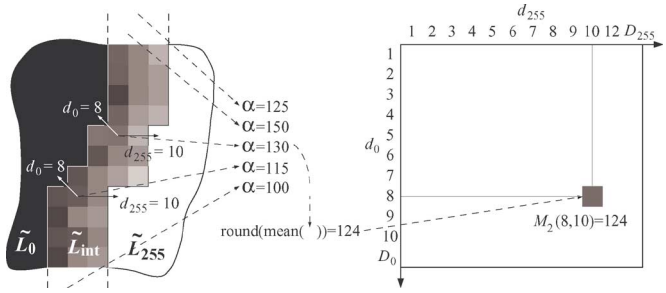


Fig. 6. Construction of the matrix M_2. Each element $M_2(j,k)$ is calculated using (5), as illustrated in this example for $j = 8$ and $k = 10$.

E. Inter Coding

Temporal correlation is exploited by introducing an additional block type denoted *inter* blocks. The algorithm is modified as follows: for each block, which is non TR and non OP, the motion compensated prediction (MCP) is first calculated. Motion vectors are estimated on co-located blocks in the binary shape. Only blocks that contain shape boundaries are utilized, i.e., if the colocated block is entirely in $\overline{L}_0$, then the nearest boundary block is utilized. Since motion estimation is performed on the binary shape, no coding overhead is required. If the result of motion compensation is good enough, i.e., if the error is within $e_{\max}$ and the alpha values in the block are not smaller than $\alpha_{\min}$, then the block is classified as *inter* and represented as the MCP, otherwise the intra algorithm is used. The VLCs in Table I are modified accordingly.

V. CODING THE INTERMEDIATE LAYER

Three alternative strategies for coding the transparency values $\alpha(\widetilde{L}_{\text{int}})$ in the intermediate layer are proposed in this section: A) the SA-DCT with modified support; B) a method based on calculation of a distance map with respect to the previous background and opaque layers; C) a method based on quantization and context-based arithmetic encoding. The three new algorithms are presented below and hereafter indicated as MS SA-DCT, DMC, and Q+BMF, respectively. Each algorithm assumes that the (coded versions of the) two previous layers L_0 and L_{255} are known.

A. SA-DCT With Modified Support (MS SA-DCT)

Since $\widetilde{L}_{255}$ is known in addition to $\widetilde{L}_0 = L_0$, the conventional SA-DCT algorithm is utilized to encode the pixels in $\widetilde{L}_{\text{int}}$ only, and not all of the pixels in $\overline{L}_0 = \widetilde{L}_{\text{int}} \cup \widetilde{L}_{255}$ as done in MPEG-4 Part 2. The advantage of this simple approach is illustrated in Fig. 7, showing that the blocking artifacts produced by MPEG-4 Part 2 in the interior of the shape are entirely avoided if the MS SA-DCT is utilized.

B. Distance Map Coder (DMC)

The transparency values $\alpha(\widetilde{L}_{\text{int}})$ are represented as a function of their pixel positions relative to previous layers $\widetilde{L}_0$ and $\widetilde{L}_{255}$. The basic idea is to replace the value of all pixels with a certain position relative to $\widetilde{L}_0$ and $\widetilde{L}_{255}$ with their average value (Fig. 6). The information to be encoded is only one average value for each possible relative position.

The description of the algorithm is organized as follows: the definitions for determining the relative position of the pixels are introduced. For the sake of clarity, a simple two parameters version of the algorithm is first described; the complete, three parameters model is then presented, followed by the description of the refinement step for encoding the residual error.

1) Position Relative to $\widetilde{L}_0$ and $\widetilde{L}_{255}$: Let $d^E(a,b)$ be the Euclidean distance between two pixels a and b. Let $\widetilde{L}_i$, $i \in \{0,255\}$ denote $\widetilde{L}_0$ or $\widetilde{L}_{255}$. Let $d_i^E(a) = \min_{b\in\widetilde{L}_i} d^E(a,b)$ be the Euclidean distance between a pixel $a \in \widetilde{L}_{\text{int}}$ and $\widetilde{L}_i$, for $i \in \{0,255\}$. Two new parameters are introduced, in part based on quantization of $d_i^E(a)$, and defined as

$$d_i(a) = \begin{cases} 9 - N_i(a), & \text{if } d_i^E(a) < 2 \\ \lceil d_i^E(a)/Q_i \rceil + 8, & \text{if } d_i^E(a) \geq 2 \end{cases} \tag{4}$$

for $i \in \{0,255\}$, where $N_i(a)$ is the number of 8-connected neighbors of the pixel a which are in $\widetilde{L}_i$, Q_i is the quantization step used for layer $\widetilde{L}_i$, and $\lceil\cdot\rceil$ is upward rounding. Using d_i instead of d_i^E allows for a more precise differentiation of the pixels on the boundary of $\widetilde{L}_{\text{int}}$, while the information for the other pixels is quantized. Let the maximum value of the distances with respect to $\widetilde{L}_0$ and $\widetilde{L}_{255}$ be denoted as $D_0 = \max_{p\in\widetilde{L}_{\text{int}}} d_0(p)$ and $D_{255} = \max_{p\in\widetilde{L}_{\text{int}}} d_{255}(p)$.

2) Simple Two Parameters Model: Let the matrix M_2 of relative distances (d_0, d_{255}) be constructed by the elements

$$M_2(j,k) = \left\lceil \mathtt{m}\left(\left\{\alpha(p) | p \in \widetilde{L}_{\text{int}}, d_0(p) = j, d_{255}(p) = k\right\}\right)\right\rfloor \quad \forall (j,k) \in \{1,2,\ldots,D_0\} \times \{1,2,\ldots,D_{255}\} \tag{5}$$

where $\lceil\cdot\rfloor$ is rounding to the nearest integer; $\mathtt{m}(\cdot)$ is the mean value of the elements in the set or 0 if the set is empty. The matrix M_2 is the information to code, since it contains the average α value for every possible pair (d_0, d_{255}).

The matrix is encoded as follows: first, if the number of pixels in $\widetilde{L}_{\text{int}}$ with a certain value of $d_0 \in [1, D_0]$ (or $d_{255} \in [1, D_{255}]$) is less than a preselected threshold (currently set to 5), then the corresponding row (or columns, respectively) of M_2 are removed. The reduced matrix, M_2', is then encoded using SA-DCT, where the support of M_2' defines the shape.

3) Reconstruction of the Layer $\widetilde{L}_{\text{int}}$: M_2' is decoded, giving the approximation $\widetilde{M_2'}$; since $\widetilde{L}_0$ and $\widetilde{L}_{255}$ are known, the indices of the rows and columns that where removed from M_2 are also known at the decoder: the values of the pruned elements are interpolated from the remaining ones (this is simply done by taking the nearest element), such that the full size matrix $\widetilde{M_2}$ is reconstructed. The transparency values are reconstructed as

$$\widetilde{\alpha}_2(p) = \widetilde{M_2}\left(d_0(p), d_{255}(p)\right), \text{ for } p \in \widetilde{L}_{\text{int}}. \tag{6}$$

Finally, a simple low-pass linear filter is applied. The filter has a 3 × 3 kernel with rows [1/10, 1/10, 1/10], [1/10, 2/10, 1/10], [1/10, 1/10, 1/10]. It is only applied within the interior of $\widetilde{L}_{\text{int}}$, i.e., only to the pixels $p \in \widetilde{L}_{\text{int}}$ with $d_0^E(p) \geq 2$ and $d_{255}^E(p) \geq 2$.

4) Full Three Parameters Model: A more precise reconstruction is obtained augmenting the model by adding a third variable: a 3-D matrix M_3 is constructed in a way similar to (5) by considering also a third variable which conveys information about the orientation of the pixels in $\widetilde{L}_{\text{int}}$ with respect to $\widetilde{L}_0$ and $\widetilde{L}_{255}$, in addition to d_0 and d_{255}. Consider the segment $\overline{ab}$ connecting the pixel a to b. Let $\theta(a,b)$ indicate the angle that this segment forms with the positive horizontal axis, so that $0 \leq \theta(a,b) < 2\pi$. Let $\theta_i(p)$ indicate the direction of the pixel $p \in \widetilde{L}_{\text{int}}$ with respect to the layer $\widetilde{L}_i$, defined as

$$\theta_i(p) = \theta(p,q), \quad 0 \leq \theta_i(p) < 2\pi \tag{7}$$

for $i \in \{0,255\}$, where q is the point in $\widetilde{L}_i$ with minimum distance from p, i.e., q is chosen such that $d^E(p,q) = d_i^E(p)$. In order to map part of the information about the directions with respect to the two previous layers to one single variable, a global direction $\theta_{0,255}(p)$ is defined as

$$\theta_{0,255}(p) = \begin{cases} \theta_0(p), & \text{if } d_0(p) \leq d_{255}(p) \\ \theta_{255}(p) + \pi, & \text{if } d_0(p) > d_{255}(p) \wedge d_{255}(p) < \pi \\ \theta_{255}(p) - \pi, & \text{if } d_0(p) > d_{255}(p) \wedge d_{255}(p) \geq \pi \end{cases} \tag{8}$$

such that $0 \leq \theta_{0,255}(p) < 2\pi$. The third variable of the model is then obtained by quantizing $\theta_{0,255}(p)$. The angular interval $[0, 2\pi)$ is partitioned in N_θ equally spaced intervals, indexed with the integers from 1 to N_θ, such that each $p \in \widetilde{L}_{\text{int}}$ can be assigned an index $\Theta(p) \in \{1,2,\ldots,N_\theta\}$. Hence, the 3-D matrix M_3 is obtained as

$$M_3(j,k,l) = \left\lceil \mathtt{m}\left(\left\{\alpha(p) | p \in \widetilde{L}_{\text{int}}, d_0(p) = j, d_{255}(p) = k, \Theta(p) = l\right\}\right)\right\rfloor \quad \forall (j,k,l) \in \{1,\ldots,D_0\} \times \{1,\ldots,D_{255}\} \times \{1,\ldots,N_\theta\}. \tag{9}$$

The matrix M_3 is encoded in the same manner as M_2. The alpha values are reconstructed in the same manner as in (6) and filtered as described above.

5) Refinement Step: For a higher quality, the transparency values reconstructed based on M_3 and thereafter filtered, denoted as $\widetilde{\alpha}_M(\widetilde{L}_{\text{int}})$, are utilized as prediction for further coding. The prediction error, shifted and scaled in order to fit in the range [0, 255], is expressed as

$$E(\widetilde{L}_{\text{int}}) = \left\lfloor \left[\alpha(\widetilde{L}_{\text{int}}) - \widetilde{\alpha}_M(\widetilde{L}_{\text{int}}) + 255\right]/2 \right\rfloor \tag{10}$$

and encoded with MS SA-DCT giving $\widetilde{E}(\widetilde{L}_{\text{int}})$. The final reconstructed alpha levels in $\widetilde{L}_{\text{int}}$ are given by

$$\widetilde{\alpha}(\widetilde{L}_{\text{int}}) = 2\widetilde{E}(\widetilde{L}_{\text{int}}) - 255 + \widetilde{\alpha}_M(\widetilde{L}_{\text{int}}). \tag{11}$$

The current implementation of the DMC only utilizes intra frame correlation. Utilizing temporal redundancies should improve coding efficiency, since, e.g., it is reasonable to expect high correlation between matrices obtained from consecutive alpha plane frames.

C. Quantization and Context-Based Coding (Q+BMF)

The alpha levels are quantized in the spatial domain and then coded with context-based arithmetic encoding. The idea was in-

TABLE III
LOSSLESS CODE LENGTHS (BYTES/FRAME) OF THE PROPOSED DSLSC INTER COMPARED TO MPEG-4 CAE AND DSLSC INTRA [3], ON SDTV RESOLUTION BINARY SHAPE SEQUENCES. PERCENTAGE GAINS WITH RESPECT TO MPEG-4 CAE INTER ARE ALSO PROVIDED

sequence, layer, frames	CAE intra	DSLSC intra	CAE inter	*DSLSC inter*
akiyo, 1, 1-300	385	211	258	**120** (-53%)
weather, 1, 1-300	425	261	260	**159** (-39%)
bream, 1, 1-300	964	381	538	**317** (-41%)
sean, 1, 2-301	569	361	476	**250** (-47%)
news, 3, 1-300	752	526	228	**170** (-25%)
coast guard, 1, 1-300	348	282	**94**	118 (26%)
stefan, 1, 1-300	226	144	210	**117** (-44%)
children, 1 (kids), 1-300	906	674	731	**528** (-28%)
td, 1, 11-124	237	207	228	**192** (-16%)
td, 2, 1-78	41	38	**4**	19 (375%)
td, 3, 8-131	1213	565	1157	**525** (-55%)
td, 4, 1-78	121	110	122	**110** (-10%)
td, 5 (robot), 1-131	563	390	468	**338** (-28%)
td, 6 (explosion), 1-29	310	79	300	**89** (-70%)
td, 7, 1-105	262	233	**170**	172 (1%)
td, 8 (rain), 1-169	3380	1643	3167	**1538** (-51%)
td, 9, 1-15	871	677	876	**649** (-26%)
td, 10, 1-64	949	736	934	**656** (-30%)
average	701	414	518	**305** (-41%)

troduced in [17] and refined in [1] by using adaptive quantization and more efficient coding. This approach is here further extended by integrating it into the proposed three layer architecture: hence, it applies only to the pixels in $\widetilde{L}_{\mathrm{int}}$; the quantization is the same as implemented in [1], i.e., a scalar quantizer minimizing the mean square error; more efficient context-based arithmetic encoding is obtained by using the BMF coder [19]. The BMF supports both lossless and nearlossless coding, and achieves close to state of the art lossless coding on different classes of gray scale images (some results for natural images are reported in [6]). Inter coding is achieved by sharing the statistics of the arithmetic encoder over a number of consecutive frames. The version utilized in this paper was modified in order to support the functionality of coding only the pixels in $\widetilde{L}_{\mathrm{int}}$ [20].

D. Comments on Complexity

The focus of the presented work is on high coding efficiency. A few comments are made about the complexity. A rough evaluation of the complexity of DSLSC intra was given in [3], indicating that fast implementations should be possible. For DSLSC inter, the most obvious complex part is the motion estimation. In practical video encoders, an option would be to re-utilize the freely available texture motion information. This also applies to the opaque layer coder. For the intermediate layer, the MS SA-DCT has the same order of complexity as the SA-DCT of MPEG-4, and the two alternative methods, DMC and Q+BMF, do not utilize motion compensation.

VI. RESULTS

The results for lossless shape coding are summarized in Tables III and IV, for standard TV (SDTV) resolution material (720×486 pixels, 30 Hz) and for lower resolutions (QCIF, CIF, NTSC SIF), respectively. Since the SDTV interlaced sequences *kids*, *weather*, *sean*, *bream*, and layers 9 and 10 of the sequence *total destruction (td)* have relatively fast motion, each frame is

TABLE IV
LOSSLESS CODE LENGTHS (BYTES/FRAME) OF THE PROPOSED DSLSC INTER COMPARED TO MPEG-4 CAE AND DSLSC INTRA [3], FOR LOW RESOLUTION BINARY SHAPE SEQUENCES. PERCENTAGE GAINS WITH RESPECT TO MPEG-4 CAE INTER ARE ALSO PROVIDED

seq., format, frames	CAE intra	DSLSC intra	CAE inter	*DSLSC inter*
kids, SIF, 1-100	255	224	217	**188** (-13%)
weather, QCIF, 1-100	65	60	40	**38** (-5%)
robot, SIF, 1-44	350	295	**286**	311 (8%)
logo, SIF, 1-100	382	334	**115**	160 (39%)
foreman, CIF, 1-207	184	113	168	**96** (-42%)
stefan, SIF, 1-100	93	84	96	**79** (-17%)
news, CIF, 1-300	196	132	74	**53** (-28%)
cyclamen, SIF, 1-100	498	381	345	**300** (-13%)
average	246	186	144	**120** (-17%)

encoded as two fields, i.e., by reading first the odd and then the even lines (the statistics gathered in the first field is exploited to initialize the probability estimates for coding the second field). On the tested SDTV material, DSLSC inter (first frame intra coded, all subsequent frames inter coded) produces code lengths 41% smaller than those obtained using MPEG-4 CAE inter. In the sequences with very slow motion *coast guard* and *td layer 2*, the block-based motion compensation in CAE provides slightly better results. The same reduction was obtained for intra coding [3]. For the lower resolution sequences, in inter mode, the average bit rate reduction of DSLSC inter over CAE inter is 17%. In intra mode, the reduction was 24% [3]. The intra encoding parameters are given in [3]. These were optimized for intra coding, independently for the SDTV and the lower resolution material. The intra parameters were reused when obtaining the DSLSC inter results. Re-optimizing the parameters for inter coding, an improvement should be expected. In inter mode, each picture is partitioned in stripes, defined by a fixed number of rows, for calculation of the reference boundary band width, d_B (Section III-B). Stripes with 18 rows were used except for the SIF sequences, where 16 rows were used to avoid an incomplete stripe.

For coding the transparency information, lossy coding is preferable, and the results are evaluated by rate-distortion performance. The rate is given by the sum of the code lengths obtained for L_{255} and L_{int}, and the peak signal to noise ratio (PSNR) is measured within the shape of the object $\overline{L}_0 = L_{255} \cup L_{\mathrm{int}}$. In Tables V and VI, results are given for a fixed PSNR quality level for each sequence matching the PSNR results obtained by MPEG-4 Part 2 with fixed $\mathrm{QP}_\alpha = 24$. The complete rate-distortion curves for some of the test sequences are reported in Figs. 8 and 9. A subset of the SDTV test sequences from the MPEG-4 video verification model [14] is utilized, focusing on those containing an actual opaque layer, as, e.g., in Fig. 1, since this is the most common situation. The first 50 frames of each alpha plane sequence were utilized. The sequences *akiyo*, *kids*, *bream*, and *weather* are extracted from natural video (e.g., by means of blue screening), while layers five and seven of the sequence *total destruction* are computer generated. Prior to coding, a cleaning step was performed for the sequences *akiyo*, *bream* and *weather*, in order to remove the noise generated by the capturing process at frame margins, by padding the closest noise-free row/column. This was mainly done in order to facilitate the calculation of the distance map,

TABLE V
INTRA CODING RESULTS FOR TRANSPARENCY INFORMATION. RATES ARE EXPRESSED IN BYTES/FRAME. ΔP IS THE PSNR DIFFERENCE COMPARED TO MOMUSYS MPEG-4, WHERE = INDICATES $\Delta \mathrm{P} = 0$

seq.	MoMuSys QP_α=24		OPC + MS SADCT		OPC + DMC		OPC + Q+BMF		OPC only
	rate	PSNR	rate	ΔP	rate	ΔP	rate	ΔP	
akiyo	985	37.5	779	=	**549**	=	973	=	173
kids	3134	30.0	1473	+1.5	1265	=	**677**	=	8
bream	1878	33.3	1166	+0.5	870	=	**747**	=	179
weather	830	37.1	538	+0.4	**355**	=	613	=	155
td, l5	3114	31.2	2127	=	1849	=	**1127**	=	141
td, l7	1361	26.4	1085	=	(718)	-0.7	**822**	=	70
average	1884		1195		934		**826**		121
relative	0%		-37%		-50%		**-56%**		-

TABLE VI
INTER CODING RESULTS FOR TRANSPARENCY INFORMATION. RATES ARE EXPRESSED IN BYTES/FRAME. ΔP IS THE PSNR DIFFERENCE RESPECT TO THE MOMUSYS MPEG-4 CASE, WHERE = INDICATES $\Delta \mathrm{P} = 0$

seq.	MoMuSys QP_α=24		OPC + MS SADCT		OPC + DMC		OPC + Q+BMF		OPC only
	rate	PSNR	rate	ΔP	rate	ΔP	rate	ΔP	
akiyo	258	36.1	**196**	=	313	=	581	=	74
kids	2539	27.2	1356	+2.0	901	=	**349**	=	8
bream	1039	31.2	708	+0.3	579	=	**331**	=	114
weather	277	35.3	282	=	**205**	=	220	=	60
td, l5	2673	28.5	1266	+0.2	1030	=	**729**	=	130
td, l7	798	24.1	505	=	466	=	**430**	=	58
average	1264		719		582		**440**		74
relative	0%		-43%		-54%		**-65%**		-

and recognizing that noisy frame margins are not meant to be coded and transmitted.

The results for MPEG-4 Part 2 were evaluated using the *MoMuSys-FDIS-V1.0-990812* reference software (Version 2, Main Profile); rate control was disabled and the quantization parameter for the transparency data (QP_α) was set to each of the values $\{10, 12, \ldots, 30\}$, covering most of the range allowed by the standard (the maximum allowed value is $\mathrm{QP}_\alpha = 31$; the quantization parameter for the texture, QP_T was set to 31). Motion estimation was performed setting the search range to 16 and utilizing quarter pixel precision (the same parameters were used for the MS SA-DCT).

In the encoder for the opaque layer, described in Section IV and hereafter named OPC, the set of possible erosion radii is $R = \{0, 1, 1.5, 2, 3, 4, 6, 8\}$; in the classification process of each block, the approximation error $e(\rho_{\mathrm{opt}})$ is obtained by counting the percent of error pixels, restricted to the pixels in $\overline{L}_0$ with a distance, not greater than 4, from the contour of L_{255}; the default settings $e_{\max} = 5\%$ and $\alpha_{\min} = 0$ are utilized for all sequences except *kids*: for this sequence, the default settings produce good result for low rates, but for high rates, the PSNR saturates; hence, for increasing PSNR, the setting with $e_{\max} = 20\%$ and $\alpha_{\min} = 20$ was utilized. The DMC is applied using the matrix M_3 defined in (10), with $N_\theta = 2$, and the subsequent refinement step. The quantization steps utilized in (4) are $Q_0 = 1$ and $Q_{255} = 1$. The quantization parameter QP is set to each of the values {10,20,31} for encoding the reduced matrix M_3' and to {10,20,24,28,31} for encoding the residual error $E(\widetilde{L}_{\mathrm{int}})$ in the refinement step. The Q+BMF method is tested applying the adaptive quantization with 8, 16, 24, 32, and 64 reconstruction levels. The picture after quantization representing the indices to the reconstruction levels is then encoded with the modified BMF both in lossless and nearlossless mode (up to maximum index difference of 5). Different rate-distortion points are obtained for DMC and Q+BMF. Only the points of the convex-hull are reported.

In Tables V and VI, the rate-distortion performance of MoMuSys MPEG-4 with $\mathrm{QP}_\alpha = 24$ for SDTV transparency sequences is reported. The corresponding bit rates obtained by the three proposed solutions for the same (or similar) distortion value are also given. It is clear that all the proposed schemes outperform MoMuSys. Both for the intra case and the inter case, the OPC+MS SA-DCT provides average bit rate reductions close to 40% compared to MoMuSys, and it is consistently better than MoMuSys for each tested sequence (not considering the slight rate increase of less than 2% for the sequence *weather* coded in inter mode). Furthermore, blocking artifacts are avoided (Fig. 7). The OPC+DMC and OPC+Q+BMF offer even higher average reductions around 50% and 60%, respectively, but for the sequence *akiyo* in inter mode OPC+MS SA-DCT and MoMuSys are considerably more efficient: this is explained recognizing that in this sequence the intermediate layer is quite broad, hence motion compensation is advantageous. Since an exhaustive analysis of the combinations of parameters offered by the proposed techniques has not been performed, improved results might be possible. Furthermore, it can be observed in Figs. 8 and 9 that the best performance at low rates is typically obtained by OPC+DMC, while OPC+Q+BMF is the best at higher rates. More rate-distortion curves are reported in [2].

The coding benefit of the proposed transparency coding techniques on the overall rate-distortion performance is illuminated by an initial experiment. The first 50 frames of the SDTV sequences *kids* and *robot* (td, l5) are considered, applying inter mode coding. The overall rate is $r = r_S + r_\alpha + r_T$, where the terms indicate the rates utilized for shape, transparency and texture, respectively. The distortion of one VO is measured by the PSNR on the composed scene $Y_C(x, y)$, defined in (1), averaging over $(x, y) \in \overline{L}_0$ and assuming that the background object is perfectly reconstructed. For simplicity only lossless shape coding is considered, as it was suggested by the operational rate-distortion optimization in [26] that lossy shape coding is useful for low rates, while for medium to high rates lossless shape coding should be utilized. We compare with optimal settings of QP_α and QP_T for the MoMuSys MPEG-4 coder, obtained as points of the convex hull of the operational rate-distortion curve. An optimal combination with $\mathrm{QP}_\alpha = 30$, $\mathrm{QP}_T = 9$ was found for the sequence *kids*, resulting in $r_S = 832$, $r_\alpha = 1719$, $r_T = 4858$ bytes/frame and $\mathrm{PSNR} = 30.2$ dB. The OPC+Q+BMF algorithm encodes the transparency with less distortion (1 dB) at $r_\alpha = 326$ bytes/frame [Fig. 9(a)], which corresponds to a saving of 19% of the total rate. For the sequence *robot* an optimal combination with $\mathrm{QP}_\alpha = 30$, $\mathrm{QP}_T = 6$ was found for MoMuSys, resulting in $r_S = 493$, $r_\alpha = 1902$, $r_T = 3901$ bytes/frame, and $\mathrm{PSNR} = 38.3$ dB. Utilizing OPC+Q+BMF, r_α was reduced to 876 bytes/frame, which gives

(a) (b) (c)

Fig. 7. Transparency values ($L_{255} \cup L_{\text{int}}$) of the first frame of *total destruction*, layer 7 (a 70 × 70 pixels portion is showed): (a) original, coded with OPC ($e_{\max} = 5\%$, $\alpha_{\min} = 255$) and (b) MS SA-DCT with $\text{QP}_\alpha = 30$, (c) coded by MPEG-4 part 2 with $\text{QP}_\alpha = 30$. Blocking artifacts are observed in the MPEG-4 case. The OPC+MS SA-DCT achieves $\text{PSNR} = 25.0$ dB using 724 bytes. For MPEG-4 the code length is 1152 bytes and the PSNR is 24.5 dB.

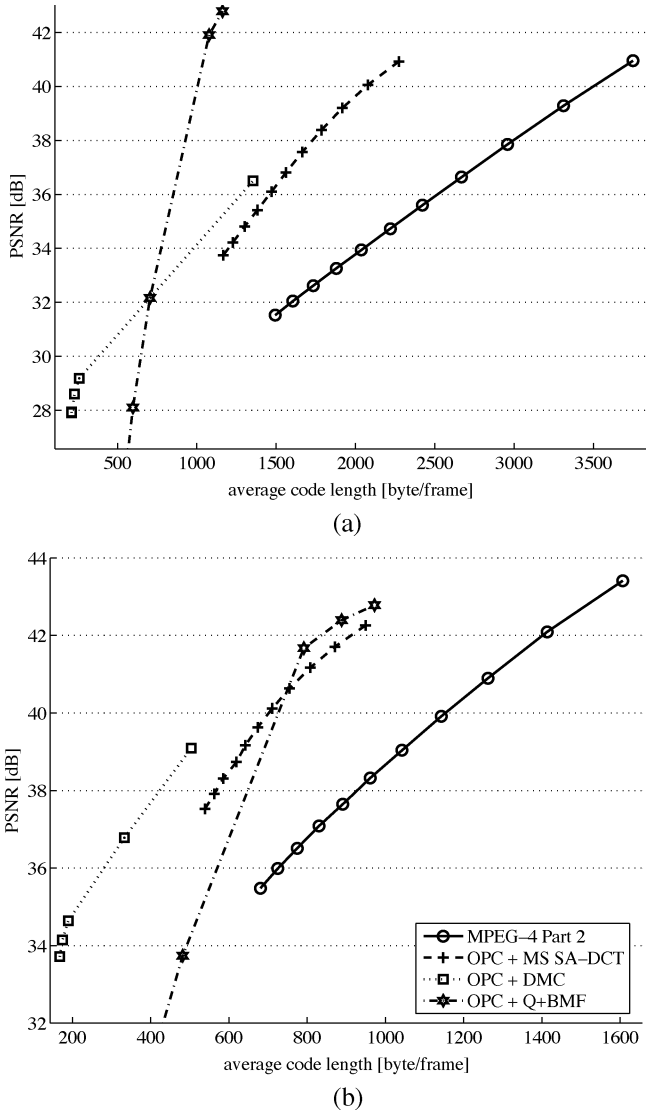


Fig. 8. Proposed methods for coding transparency data compared to MPEG-4 part 2: sequences (a) bream and (b) weather coded in intra mode.

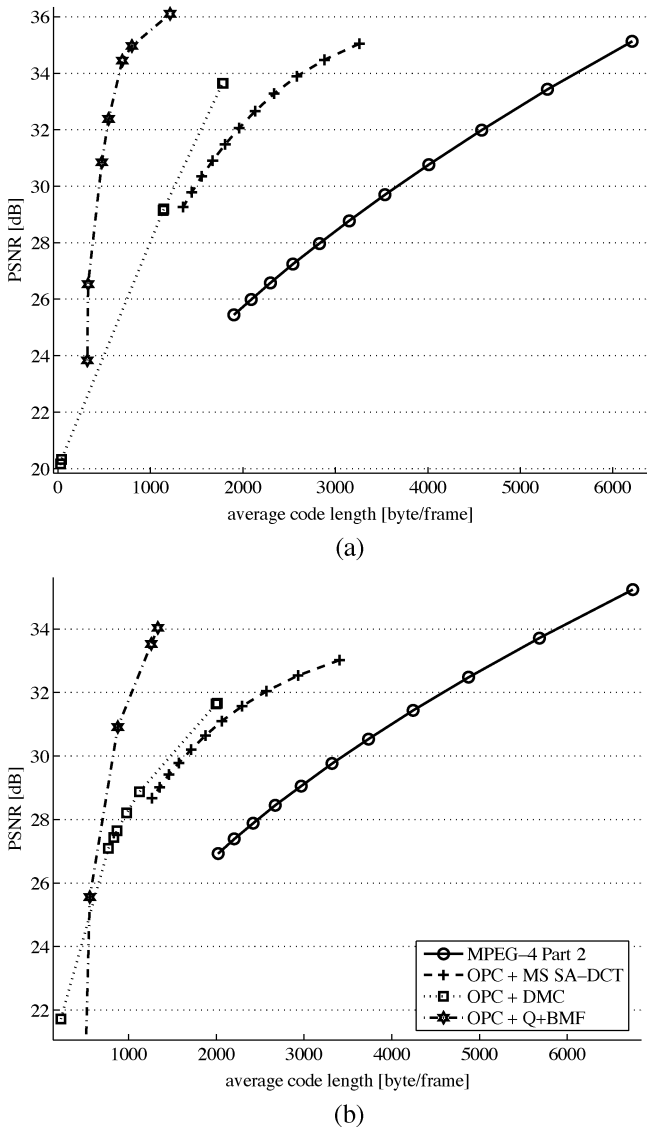


Fig. 9. Proposed methods for coding transparency data compared to MPEG-4 part 2: the sequences (a) kids and (b) robot coded in inter mode.

a saving of 18% of the total rate, together with a smaller distortion on the transparency information [4 dB, Fig. 9(b)]. It should be remarked that coding the shape with DSLSC would reduce r_S considerably. Further, if the texture was encoded with high ef-

ficiency, e.g., by an H.264/MPEG-4 Part 10 encoder eventually modified in order to process textures of arbitrary shape, r_T could also be considerably reduced. In this setting, the benefit of the proposed techniques for coding of transparency data would relatively be even more significant. Furthermore, the importance of transparency information for subjective quality is probably underestimated by PSNR as is the case for shape coding in [26], since a poorly reconstructed alpha plane does not assure a realistic representation of object boundaries.

VII. Conclusion

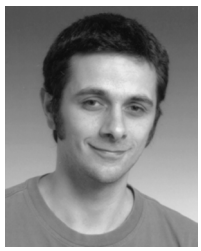
A new architecture for encoding both the shape and the transparency information in gray-level alpha planes for object-based video was proposed: the data is segmented in binary shape layer, opaque layer and intermediate layer, whereas in MPEG-4 Part 2 there is only one transparency layer for pixels located within the binary shape. The binary shape is encoded with the proposed extended DSLSC scheme. The DSLSC bilevel image coder [3] is employed for intra mode coding. The DSLSC image coder was extended to inter mode coding. As part of this, a novel approach of exploiting the temporal correlation was introduced utilizing the fact that variations between consecutive frames occur mostly along the border of the video object [4]. The opaque layer is encoded with a new strategy based on block-based partitioning and morphological erosion of the previously encoded binary shape layer. Several techniques are proposed for coding the intermediate layer: A new algorithm based on calculation of the distance map relative to the background and opaque layers, a modification of MPEG-4 SA-DCT which takes into account the presence of the opaque layer, and a simple scheme based on adaptive quantization and context based arithmetic encoding. The proposed algorithms outperform the corresponding MPEG-4 Part 2 tools. On the tested SDTV material, the new DSLSC inter produced bit rate reductions of 41% compared to MPEG-4 CAE inter, and average savings within 37%–65% compared to MPEG-4 SA-DCT were obtained by the proposed techniques for coding transparency. The presented techniques could be utilized in future object-based video encoders.

Acknowledgment

The authors would like to thank D. Shkarin for providing the modified version of the BMF image coder that was utilized in this work.

References

[1] S. M. Aghito and S. Forchhammer, "Context based coding of quantized alpha planes for video object," in *Proc. MMSP*, Dec. 2002, pp. 101–104.
[2] S. M. Aghito, "Algorithms for object-based video coding," Ph.D. dissertation, COM-DTU, Lyngby, Denmark, 2006.
[3] S. M. Aghito and S. Forchhammer, "Context based coding of bi-level images enhanced by digital straight line analysis," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2120–2130, Aug. 2006.
[4] S. M. Aghito and S. Forchhammer, "Efficient coding of binary shape sequences for object based video," presented at the Int. Workshop VLBV, Sep. 2005.
[5] P. J. Ausbeck, Jr., "The piecewice-constant image model," *Proc. IEEE*, vol. 88, no. 11, pp. 1779–1789, Nov. 2000.
[6] K. U. Barthel, "Verlustlose Bildkompression (lossless image compression)," *Inf. Technol.*, vol. 45, no. 5, pp. 247–255, Oct. 2003.
[7] L. Dorst and W. M. Smeulders, "Discrete representation of straight lines," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 4, pp. 450–463, Jul. 1984.
[8] *Coded Representension of Picture and Audio Information—Progressive Bi-Level Image Compression*, ISO/IEC Int. Std. 11544, 1993.
[9] *Coded Representension of Picture and Audio Information—Lossy/Lossless Coding of Bi-Level Images (JBIG2)*, ISO/IEC Int. Std. 14492, 2000.
[10] *Information Technology—Coding of Audio-Visual Objects—Part 2: Visual*, ISO/IEC Int. Std. 14496-2, 1999.
[11] *Information Technology—Coding of Audio-Visual Objects—Part 10: Advanced Video Coding*, ISO/IEC Int. Std. 14496-10, 2005.
[12] A. K. Katsaggelos, L. P. Kondi, F. W. Meier, J. Ostermann, and G. M. Shuster, "MPEG-4 and rate-distortion-based shape-coding techniques," *Proc. IEEE*, vol. 86, no. 6, pp. 1126–1154, Jun. 1998.
[13] V. A. Kovalevsky, "New definition and fast recognition of digital straight segments," in *Proc. Int. Conf. Pattern Recognition*, 1990, pp. 31–34.
[14] W. Li, J.-R. Ohm, M. van der Shaar, H. Jiang, and S. Li, MPEG-4 Video Verification Model Version 18.0, ISO/IEC JTC1/SC29/WG11 Doc. N3908, 2001.
[15] P. Nunes, F. Marques, F. Pereira, and A. Gasull, "A contour-based approach to binary shape coding using a multiple grid chain code," *Signal Process.: Image Commun.*, vol. 15, no. 7–8, pp. 585–599, 2000.
[16] J. Ostermann and A. Vetro, "2D shape coding," in *Document and Image Compression*. Boca Raton, FL: Francis Books, 2006.
[17] L. Piron and M. Kunt, "Differential coding of alpha planes with adaptive quantization," *ITG Fachberichte*, pp. 713–718, 1997.
[18] A. Rosenfeld and R. Klette, "Digital straightness," *Electronic Notes Theoret. Comput. Sci.*, vol. 46, pp. 1–32, 2001.
[19] D. Shkarin, BMF Version 2.0 [Online]. Available: http://www-lat.compression.ru/ds/bmf_2_hz.rar
[20] D. Shkarin, *personal communication.* 2005.
[21] G. M. Schuster and A. K. Katsaggelos, "Motion compensated shape error concealment," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 501–510, Feb. 2006.
[22] L. D. Soares and F. Pereira, "Temporal shape error concealment by global motion compensation with local refinement," *IEEE Trans. Image Process.*, vol. 15, no. 6, pp. 1331–1348, Jun. 2006.
[23] M. Van Der Schaar, D. S. Turaga, and T. Stockhammer, *MPEG-4 Beyond Conventional Video Coding: Object Coding, Resilience and Scalability*. San Rafael, CA: Morgan & Claypool, 2006.
[24] A. Vetro, T. Haga, K. Sumi, and H. Sun, "Object-based coding for long-term archive of surveillance video," in *Proc. Int. Conf. Multimedia and Expo ICME*, 2003, vol. 2, pp. 417–420.
[25] H. Wang, G. M. Schuster, A. K. Katsaggelos, and T. N. Pappas, "An efficient rate-distortion optimal shape coding approach using a skeleton-based decomposition," *IEEE Trans. Image Process.*, vol. 12, no. 10, pp. 1181–1193, Oct. 2003.
[26] H. Wang, G. M. Schuster, and A. K. Katsaggelos, "Rate-distortion optimal bit allocation for object-based video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 9, pp. 1113–1123, Sep. 2005.
[27] H. Wang, F. Zhai, Y. Eisenberg, and A. K. Katsaggelos, "Cost-distortion optimized unequal error protection for object-based video communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 12, pp. 1505–1516, Dec. 2005.
[28] L. Zhou and S. Zahir, "A novel shape coding scheme for MPEG-4 visual standard," in *Proc. 1st Inf. Conf. Innovative Computing, Information and Control*, Aug. 2006, vol. 3, pp. 585–588.



Shankar Manuel Aghito (M'06) received the M.S. degree in telecommunications engineering from the University of Padova, Padova, Italy, in 2002, and the Ph.D. degree from the Technical University of Denmark, Lyngby, in 2006.

He currently holds a postdoctoral research position at the Research Center COM, Technical University of Denmark. His primary research interests include image and video compression.

Søren Forchhammer (M'03) received the M.S. degree in engineering and the Ph.D. degree from the Technical University of Denmark, Lyngby, in 1984 and 1988, respectively.

Currently, he is an Associate Professor with the Research Center COM, Technical University of Denmark, where he has been since 1988. His main interests include source coding, image and video compression, 2-D fields, and visual communications.